

Design Patterns

The Singleton Pattern:

In the server application:

- We need to have only one instance of `EmployeeCache`, and you can get the instance by `getInstance()` method.
- We need to have only one instance of `BasicConnectionPool`, and you can get the instance by `getInstance()` method.

In the client application:

- We need to have only one instance of `AppDataManager`, and you can get the instance by `getInstance()` method.
- We need to have only one instance of `AppDBHelper`, and you can get the instance by `getInstance()` method.

Command and Query Responsibility Segregation (CQRS) pattern:

- `EmployeeFindAllQuery`, to get all employees.
- `EmployeeFindByIdQuery`, to get single employee by id if exists, `Optional.empty()` otherwise.
- `EmployeeInsertQuery`, to insert a new employee.
- `EmployeeUpdateByIdQuery`, to update an employee by id.
- `EmployeeDeleteByIdQuery`, to delete an employee by id.

(MVC) design pattern:

- `MainView` act as View that presents the model's data to the user.
- `MainController` act as Controller that exists between the view and the model
- `DataManager` act as Model that contains only the pure application data.